

Association for Information Systems AIS Electronic Library (AISeL)

AMCIS 1997 Proceedings

Americas Conference on Information Systems
(AMCIS)

8-15-1997

A Process Model of Initial Representation Formation: Building Intelligent Agents for Problem Structuring

Dolphy Abraham
Loyola Marymount University

Follow this and additional works at: <http://aisel.aisnet.org/amcis1997>

Recommended Citation

Abraham, Dolphy, "A Process Model of Initial Representation Formation: Building Intelligent Agents for Problem Structuring" (1997). *AMCIS 1997 Proceedings*. 25.
<http://aisel.aisnet.org/amcis1997/25>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 1997 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

A Process Model of Initial Representation Formation: Building Intelligent Agents for Problem Structuring

Dolphy Abraham
College of Business Administration
Loyola Marymount University

Abstract

Problem structuring has been identified as a phase of the decision making process that has not been studied in detail. It is necessary to better understand this process in order to enhance the quality of decision support systems. Initial Representations of problems are a starting point in the problem structuring process. This paper describes a process model of initial representation formation and how the model can be used to create an intelligent agent in the context of a DSS.

Introduction

Problem structuring has been identified as a phase of the decision making process that has not been addressed sufficiently in the Decision Support Systems literature (Konsynski, Stohr and McGee, 1992). The process of problem structuring can be viewed as part of the intelligence phase (Simon, 1960) of decision making. Simon (1973) suggests that problem solvers impose structure on the problem as they work on it. He used the term *initial representation* to denote a model that the problem solver creates using information about the problem on which he/she is working. The initial representation, he suggested, served to constrain the problem space and hence helped in arriving at a solution. Therefore, studying the process by which a problem solver creates an initial representation of an ill-structured problem would help us understand the process of problem structuring.

A model of the initial representation formation process would serve to support the decision making process better especially when the decision maker is faced with a situation that is new to him/her. The model implemented within the DSS would be able to create an initial structure that would be a starting point for the decision maker. In this paper, we present a process model that describes the formation of an *initial representation* of an ill-structured problem. This process model is the first step in a research stream that aims to address the gap in the DSS literature.

Process Model of Initial Representation

The process model was developed by collecting verbal protocols from a decision maker working on a Local Area Network planning problem. The Initial Representation Structure and process model described below are based on the analysis of those protocols.

Local Area Network (LAN) planning falls into the category of unstructured problems because it typically involves a situation where all the factors about the problem are not known ahead of time. Issues such as the desired solution have to be determined with a significant amount of analysis. In many cases, the requirements may not be able to be specified before the problem solving starts. The network analyst has to first identify the business problems to be solved and then define the network requirements.

The Initial Representation Structure

The process model uses a data/knowledge structure to represent information within the DSS environment. The Initial Representation (IR) Structure is based on Groen and Patel's (1988) description of a well-structured problem. During IR formation, information necessary to fit the slots is either retrieved from long-term memory or requested from external sources (including other people). There are four slots in the IR structure:

1. The Problem Definition containing information about the current state and the desired state of the world. The states are described in terms of the problem context and not in technical terms. This can be viewed as an external view of the problem.
2. The Goal State identifying a goal state retrieved from memory during the IR formation process. This slot contains a technical representation of the goal state and can be viewed as an internal view of the problem.
3. The Solution Method(s) contains the information about methods or tools that the problem solver knows (i.e., retrieves from memory) will help him/her get to the goal state.
4. The Constraint(s) are information and assumptions collected from both internal and external sources which restrict the problem solving process.

The IR structure as described above is intended to work as a problem representation beyond just the initial phase of problem solving. Therefore, it is not necessary for all the slots to be filled up before the initial representation process is completed. Information may be entered in the slots even after the initial representation process is complete.

The Process Model

The process model consists of two parts: the *Initial Representation Processor* (IRP) and the *DONE? predicate*. The IRP is a collection of functions that accomplishes the tasks observed in the subject's behavior. In brief, the IRP takes information from an external source and puts each piece in the appropriate slot of the IR structure. When needed, it requests information from an external source. The IR structure is checked to see if there is sufficient information to stop the process, and if so, the IR structure is then available for use by the rest of the DSS.

The IRP takes information from both external and internal sources and fits them into the appropriate slots in the IR frame. In order to do this, the IRP has to first, evaluate the information and then classify it appropriately. When the information is incomplete, then additional information must be requested. Additional information can be retrieved from memory or from external sources using a function called ASK. This function will formulate the information request as needed. In addition, there is a function called LISTEN that will sense the information available in the electronic environment of the DSS and pick those items that are relevant. In this paper, we focus on the IRP and the DONE? predicate.

The Initial Representation Processor consists of three groups of functions. These are:

1. Process Information Received: The goal of this group of functions is to determine how the information can be used. The functions in this group are as follows:

Classification

Relevancy

Redundancy

Consistency

The information that is received can be considered to be in a stack. Each item is processed from the top of the stack. If follow-up information is requested then new information is put on top of the stack and the processing continues. When the stack is empty, the Process Information Received group ceases to act and the Generate Information Request process is triggered to pass the IR structure to the DONE? predicate.

2. Generate Information Request: This group is triggered for two reasons. First, it is triggered when the Process Information Received group has worked through all the information received. In this case, the IR structure has to be checked for completeness and passed onto the DONE? predicate. In the second case, this group is triggered when some information is requested for a specific purpose. This could be to clarify some incomplete piece of information, or to resolve the problem of conflicting pieces of information. The four functions in this group are as follows:

Check for Completeness

Determine Information Required

Retrieve from Memory

Prompt ASK process

The IR frame is determined to be complete when there is no more information in the stack that needs to be clarified and when all the relevant pieces of information as well as heuristics have been retrieved from long term memory. It is then passed on to the DONE? predicate to check if the process of Initial Representation Formation is complete.

3. Fill Incomplete Slots: The third group of functions deals with situations when IR frame does not meet the conditions of the DONE? predicate. The conditions of the DONE? predicate are described below. The functions needed are as follows:

Identify incomplete Slot

Trigger Generate Information Request

Make an assumption

This group of functions are triggered when the IR frame does not contain enough information to meet the conditions of the DONE? predicate. Like the DONE? predicate these functions may be triggered several times during the course of the initial representation formation process. In the next section, the DONE? predicate is described in more detail.

The DONE? Predicate is intended to signal when the IR processor can stop working and the problem solver take over entirely, i.e., when the process of initial representation formation is complete. The predicate can be represented as a function, which checks slots of the Initial Representation frame for certain conditions that have to be present for the problem to be structured. The predicate was based on the problem structuring behavior of the subject where he stopped asking for information and proceeded to come up with a solution. By analyzing the information he had either requested and which was provided by the experimenter, the conditions were determined.

Based on the subject's verbalizations it was inferred that the knowledge of current and desired states may be viewed as a necessary condition for initial representation formation. Without this information, it would not be possible to define the problem. In addition, the goal state i.e., the internal description or the problem solver's description of the problem cannot be determined without knowing this information. Both the goal state as well as the problem definition have to be known before there is sufficient information to solve the problem. Once the goal state known, then the appropriate solution methods can be chosen and problem solved. However, in order to solve the right problem i.e., the solution matches the desired state, the desired state has to be known beforehand.

Ongoing Work

Intelligent Agents have been shown to be useful in the context of DSS especially, to provide the ability for the DSS to adapt to the changing conditions under which the decision is made (Abraham and De', 1996). Intelligent agents provide the programming tool that will enable the process model described above to work in the context of a DSS.

In particular, the intelligent agent would use the model described above to keep track of changes in the DSS environment, by using the component functions of the process model, it will be able to *sense* changes, *communicate* with other agents or with the decision maker, *reason* using domain specific heuristics and *learn* by continually developing the Initial Representation structure. This agent would be one among several types of agents that can potentially be used in a DSS. By implementing this model, using the intelligent agent paradigm, DSS can be made to support the decision making process better, especially the early stages involving problem structuring.

References

Abraham D. M and R. De' (1996). Using Intelligent Agents in Decision Support Systems, *Proceedings of the First INFORMS Conference on Information Systems and Technology*, May 5-8, 1996, Washington, D.C.

Groen, G. and V.L. Patel (1988). The relationship between comprehension and reasoning in medical expertise. In M. Chi, R. Glaser, and M. Farr (Eds.), *The Nature of Expertise*, pp. 287-310, Lawrence Erlbaum Associates, Hillsdale, NJ.

Konsynski, B., E.A. Stohr, and J.V. McGee (1992). Review and Critique of DSS. In E. Stohr and B. Konsynski (Eds.), *Information Systems and Decision Processes*, Chapter 1, pp. 7-26, IEEE Computer Society Press, Los Alamitos, CA.

Simon, H. (1960). *The New Science of Management Decision*. Prentice Hall, Englewood Cliffs, NJ.

Simon, H. (1973). The Structure of Ill-structured Problems. *Artificial Intelligence*, 4, 181-201.